

# 基于环境建模与自适应窗口的机器人路径规划

仲训昱 彭侠夫 缪孟良

(厦门大学 信息科学与技术学院, 福建 厦门 361005)

**摘要:** 为解决未知环境下移动机器人的实时路径规划问题, 提高路径的综合性能, 采用多约束局部环境建模方法, 综合考虑了各行进方向上的可通行性、安全性、运动平稳性和目标引导性等因素; 在环境建模基础上, 将 Bug 算法与基于滚动窗口的路径规划相结合, 提出一种改进的路径规划算法, 该算法为机器人设计了趋向目标行为和克服局部极小的沿墙行走行为, 其滚动窗口的大小能够根据环境进行自适应动态调整. 仿真结果表明, 提出的环境建模及路径规划方法具有较好的避障能力和环境适应性.

**关键词:** 移动机器人; 路径规划; 环境建模; 自适应调整; 滚动窗口

**中图分类号:** TP24      **文献标识码:** A      **文章编号:** 1671-4512(2010)06-0107-05

## Planning of robot paths through environment modelling and adaptive window

Zhong Xunyu Peng Xiafu Miao Mengliang

(School of Information Science and Technology, Xiamen University, Xiamen 361005, China)

**Abstract:** In order to plan real-time paths for mobile robots and improve the synthetical performance of the robot's trajectory, a multi-constrained local environment-modeling method was used, in which the traversability, security and movement smoothness were taken into consideration. On the basis of environment modeling, a improved path planning algorithm, including move to goal behavior and wall following behavior based on adaptive window, is proposed by combining Bug algorithm and rolling path planning. Simulation results show that the proposed methods have good obstacle avoidance capabilities and environment adaptability.

**Key words:** mobile robot; path planning; environment modeling; adaptive adjusting; rolling window

针对未知环境下的移动机器人实时路径规划, 文献[1~3]借鉴预测控制的广义原理, 提出了基于滚动窗口的滚动规划算法, 但是该方法假定所有障碍物为凸型, 只能保证在凸形障碍物环境中的可达性和安全性, 当障碍物形状过于复杂或特殊时机器人就会陷入死锁或振荡状态<sup>[4]</sup>, 单靠实时探测的局部环境信息无法进一步规划路径. 并且滚动规划中采用的窗口的大小是固定的, 在复杂多变的环境中适应性不够好<sup>[4]</sup>, 而随着窗口的增大, 每一次滚动规划中的局部规划时间会迅速增长, 这样总的规划时间也就显著增加了<sup>[5]</sup>. 此外, 为了提高机器人的环境适应性, 路径规划还应

该综合考虑外界约束条件, 以及路程、时间、安全性和平稳性等指标选择最佳路径, 而不是简单地选择从起始点到目标点的最短无碰路径<sup>[6~8]</sup>, 本文在综合 Bug 算法<sup>[9~11]</sup>和滚动规划算法的基础上, 提出一种基于自适应窗口的路径规划算法. 设计了机器人的趋向目标行为和沿墙行走行为.

### 1 局部环境建模与分析

#### 1.1 机器人局部环境

如图 1 所示, 机器人用激光传感器来感知障碍物的信息, 每一个障碍物探测点相对于机器人

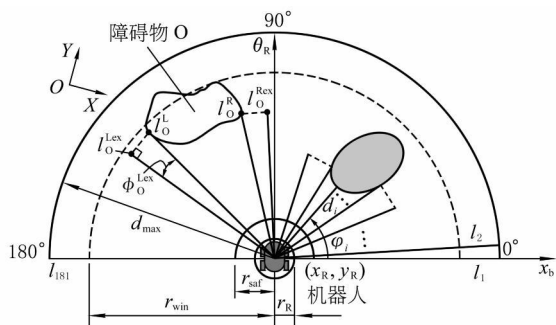


图 1 机器人局部环境示意图

的位置可由激光传感器直接得出. 图中:  $(x_R, y_R)$  为机器人的中点位置坐标;  $\theta_R$  为机器人方向;  $r_R$  为机器人半径. 传感器扫描范围为  $180^\circ$ , 角度扫描间隔为  $1^\circ$ , 共得到 181 个扫描点. 定义从  $x_b$  轴正向开始, 逆针对扫描点进行标记为  $l_1, l_2, \dots, l_i, \dots, l_{181}$ , 其中  $l_i$  以扫描角  $\varphi_i$  和对应的测量距离  $d_i$  来表示, 即极坐标形式

$$l_i = \{(\varphi_i, d_i) \mid \varphi_i = i - 1, d_i \leq d_{\max}\},$$

式中:  $i = 1, 2, \dots, 181$ ;  $d_{\max}$  为传感器最大测量距离 ( $d_i < d_{\max}$  时,  $(\varphi_i, d_i)$  即为一个障碍物点);  $\varphi_i$  为  $l_i$  相对于  $x_b$  轴的角度 (当  $\varphi_i = 90^\circ$  时,  $l_i$  与机器人同向), 在全局坐标系  $XOY$  中对应的角度为:  $\theta = \varphi_i + \theta_R - 90^\circ$ .

根据以上传感器扫描探测到的局部环境信息, 设计了一个动态自适应规划窗口, 其半径  $r_{\text{win}}$  可根据环境信息动态调整 ( $r_{\min} \leq r_{\text{win}} \leq d_{\max}$ ,  $r_{\min}$  大于安全半径  $r_{\text{saf}}$ ), 这样就可把  $t$  时刻测距范围内探测到的局部环境信息在半径为  $r_{\text{win}}$  的半圆窗口中表示为  $\text{win}(r_{\text{win}}(t)) = \{l_1, l_2, \dots, l_i, \dots, l_{181}\}$ , 其中扫描点  $l_i$  被重新定义为

$$l_i = \{(\theta, \rho) \mid \theta = \theta_i, \rho = \begin{cases} d_i & (d_i < r_{\text{win}}) \\ r_{\text{win}} & (d_i \geq r_{\text{win}}) \end{cases}\},$$

式中  $\rho$  为重新定义后的测量距离. 当  $l_i$  被称为序列时, 即表示一个射线方向; 当称为点时, 即为一个极坐标点. 以下对  $\text{win}(r_{\text{win}}(t))$  中  $l_i$  的可通行性、安全性、运动平稳性及目标引导性进行分析.

### 1.2 可通行性分析

定义 1 对局部窗口  $\text{win}(r_{\text{win}}(t))$  中的扫描序列  $l_i$ , 若  $\rho < r_{\text{win}}$  且  $\rho_{+1} = r_{\text{win}}$ ,  $\rho_{-1} < r_{\text{win}}$ , 则称  $l_i$  为某一障碍物的左边界线; 若  $\rho < r_{\text{win}}$  且  $\rho_{+1} < r_{\text{win}}$ ,  $\rho_{-1} = r_{\text{win}}$ , 则称  $l_i$  为某一障碍物的右边界线.

采用边界线扩展法对障碍物进行“膨胀”: 如图 1 所示, 序列  $l_i^b$  为不规则障碍物  $O$  的左边界线, 将其扩展至  $l_i^{\text{lex}}$ , 对应左边界点  $l_i^b$  的扩展点为  $l_i^{\text{lex}}$ ,  $|l_i^b l_i^{\text{lex}}| = r_{\text{saf}}$ , 且  $l_i^b l_i^{\text{lex}}$  垂直于  $l_i^{\text{lex}}$ , 则序列  $l_i^b$  和  $l_i^{\text{lex}}$  间的夹角  $\phi_i^{\text{ex}} = \arcsin(r_{\text{saf}}/\rho_i)$ , 然后把  $\phi_i^{\text{ex}}$

角度范围内所有的扫描点等同为  $l_i^b$ , 即令  $\rho = \{\rho \mid \rho = \rho_i, \theta_i^b \leq \theta \leq \theta_i^b + \phi_i^{\text{ex}}\}$ ; 同理可对障碍物  $O$  的右边界线进行膨胀.

对障碍物膨胀后, 在路径规划中机器人可视为一个不记大小的质点. 在传感器扫描的各方向上, 若存在障碍物则认为该方向不能通行, 于是定义  $l_i$  方向上的可通行代价函数为

$$f_{l_i}^{\text{trav}}(\rho) = \begin{cases} 0 & (\rho = r_{\text{win}}); \\ +\infty & (\rho < r_{\text{win}}). \end{cases}$$

定义 2 若  $f_{l_{i-1}}^{\text{trav}} = +\infty, f_{l_i}^{\text{trav}}, f_{l_{i+1}}^{\text{trav}}, f_{l_{i+2}}^{\text{trav}}, \dots, f_{l_j}^{\text{trav}}$  均为 0, 且  $f_{l_{j+1}}^{\text{trav}} = +\infty$ , 则称序列  $l_i \sim l_j$  为局部窗口  $\text{win}(r_{\text{win}}(t))$  中的一个可通行方向域,  $l_j$  和  $l_i$  分别为此方向域的左右边界. 对任意的  $l_i$ , 若  $f_{l_i}^{\text{trav}} = 0$  则其肯定属于某一可通行方向域, 记作  $F(l_i)$ , 即为  $l_i$  所在的可通行方向域, 其左右边界分别记为  $l_i^{\text{FDL}}$  和  $l_i^{\text{FDR}}$ .

### 1.3 安全性分析

对可通行  $l_i$  序列来说, 其与障碍物边界线的夹角越大, 机器人沿此方向行进的安全性就越高, 定义  $l_i$  的安全性度量函数为

$$f_{l_i}^{\text{saf}}(\theta) = \begin{cases} \min(\theta^{\text{FDL}} - \theta_i, \theta - \theta^{\text{FDR}}) & (\rho = r_{\text{win}}); \\ +\infty & (\rho < r_{\text{win}}), \end{cases}$$

式中  $\theta^{\text{FDL}}$  和  $\theta^{\text{FDR}}$  分别为  $F(l_i)$  的左右边界  $l_i^{\text{FDL}}$  和  $l_i^{\text{FDR}}$  所对应的方向角.

### 1.4 运动平稳性分析

机器人可以在  $360^\circ$  范围任意转向, 但实际应用中总期望机器人在行进过程中能够减少转向操作, 尽量沿着比较平滑的轨迹行驶, 这样有利于车体定位操作, 同时降低能量消耗.

为了将路径的平稳性反映到路径规划模型中, 引入路径平滑性因素, 定义  $l_i$  的平滑性代价函数为  $f_{l_i}^{\text{smooth}}(\theta) = |\theta_R - \theta|$ .

### 1.5 目标引导性分析

目标点  $P_G = (x_G, y_G)$  对于局部环境中任一方向序列  $l_i$  的路径引导函数为  $f_{l_i}^{\text{goal}}(\theta) = |\theta_{\text{goal}} - \theta|$ , 式中  $\theta_{\text{goal}} = \arctan[(y_G - y_R)/(x_G - x_R)]$  为目标点  $P_G(x_G, y_G)$  相对于机器人的方向 ( $\theta_{\text{goal}} \in [-\pi, \pi]$ ).

## 2 基于环境模型的实时路径规划

在环境建模和分析的基础上, 设计了一种基于自适应窗口的实时路径规划方法.

### 2.1 基于自适应动态窗口的路径规划算法

#### 步骤 1 系统参数初始化

步骤 2 对当前规划窗口  $\text{win}(r_{\text{win}}(t))$  内的环境信息进行更新。

步骤 3 趋向目标运动 (MtG), 直到满足以下条件之一: a. 到达目标点, 停止运动; b. 发现局部极小, 转步骤 4。

若目标点位于可通行域中, 则直接以目标点作为当前的子目标, 否则利用启发式方法确定当前  $\text{win}(r_{\text{win}}(t))$  内的即时子目标; 若无法确定即时子目标, 机器人根据目标方向自转  $90^\circ$  进行探测 (即以  $(\theta_R \pm \pi/2, 0)$  为子目标); 最后根据环境信息自适应调整规划窗口  $\text{win}(r_{\text{win}}(t))$  的半径  $r_{\text{win}}(t+1)$ , 转步骤 2。

步骤 4 沿墙行走 (WF), 直到满足以下条件之一: a. 到达目标点, 停止运动; b. 满足退出条件, 转步骤 3; c. 若沿墙行走的距离大于设定值仍未退出, 则认为目标不可达, 停止运动。

根据目标点位置, 选择 follow-left-wall (沿左墙行走) 或 follow-right-wall (沿右墙行走); 若目标点位于可通过域中, 则直接以目标点作为当前的子目标, 否则根据选择的沿墙走方式, 选择当前  $\text{win}(r_{\text{win}}(t))$  内的即时子目标; 最后根据环境信息调整规划窗口的半径  $r_{\text{win}}(t+1)$ , 转步骤 2。

2.2 机器人的趋向目标行为

2.2.1 即时子目标的确定

由于目标点  $P_G$  不一定包含在  $\text{win}(r_{\text{win}}(t))$  的可通行方向域内, 因此必须在每次局部规划窗口确定机器人的期望路径方向, 即生成局部即时子目标, 它可看作  $P_G$  在  $\text{win}(r_{\text{win}}(t))$  内的映射。下面讨论 MtG 行为中即时子目标的产生方式。

对于  $\text{win}(r_{\text{win}}(t))$ , 若  $P_G \notin \text{win}(r_{\text{win}}(t))$ , 则利用启发式函数  $f(l_i) = g(l_i) + h(l_i)$  来选取使  $f(l_i)$  最小的扫描点作为子目标 (记作  $l_{gh}$ ), 即

$$l_{gh} = \arg \min_{l_i \in \text{win}(r_{\text{win}}(t))} (f(l_i)),$$

$$f(l_i) = g(l_i) + h(l_i),$$

式中:  $g(l_i)$  为机器人从当前位置向  $l_i$  方向行进的代价;  $h(l_i)$  为从  $l_i$  方向行进到目标点方向的代价。根据上节环境建模的分析, 定义  $g(l_i) = f_{l_i}^{\text{trav}}$ ,  $h(l_i) = f_{l_i}^{\text{goal}}$ 。确定  $l_{gh}$  后, 在其可通过方向域  $F(l_{gh})$  中求安全性子目标  $l_{\text{safe}}$  和平稳性子目标  $l_{\text{smooth}}$ :

$$\begin{cases} l_{\text{safe}} = \arg \min_{l_i \in F(l_{gh})} (f(l_i)) \\ \quad (f(l_i) = f_{l_i}^{\text{safe}}(\theta_i)); \\ l_{\text{smooth}} = \arg \min_{l_i \in F(l_{gh})} (f(l_i)) \\ \quad (f(l_i) = f_{l_i}^{\text{smooth}}(\theta_i)). \end{cases}$$

至此, 根据不同的性能指标要求得到位于可通过

方向域  $F(l_{gh})$  中的 3 种子目标点:  $l_{gh} = (\theta_{gh}, r_{\text{win}})$ ,  $l_{\text{safe}} = (\theta_{\text{safe}}, r_{\text{win}})$  和  $l_{\text{smooth}} = (\theta_{\text{smooth}}, r_{\text{win}})$ 。如图 2 所示, 综合考虑安全性和运动平稳性约束, 对子目标方向进一步优化选择, 最终即时子目标

$$P_{\text{sub}} = \{(\theta_{\text{sub}}, \rho_{\text{sub}}) \mid \theta_{\text{sub}} = k_{gh}\theta_{gh} +$$

$$k_{\text{safe}}\theta_{\text{safe}} + k_{\text{smooth}}\theta_{\text{smooth}}, \rho_{\text{sub}} = r_{\text{win}}(t)\},$$

其中 3 种子目标点的权值系数  $k_{gh}$ ,  $k_{\text{safe}}$ ,  $k_{\text{smooth}}$  大于零, 且须满足  $k_{gh} + k_{\text{safe}} + k_{\text{smooth}} = 1$ 。

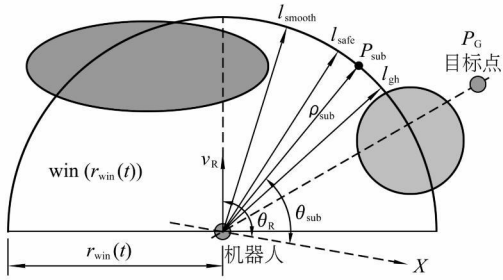


图 2 子目标优选示意图

2.2.2 规划窗口自适应调整

动态窗口  $\text{win}(r_{\text{win}}(t))$  中总的可通行方向域记为  $\varphi_F$ , 定义

$$\varphi_F = \left\{ \sum_{i=1}^{181} k_F \sin \varphi_i \mid \rho_i = r_{\text{win}} \right\}.$$

在 MtG 行为中,  $\text{win}(r_{\text{win}}(t))$  的调整原则为: 当  $\varphi_F$  较大时, 可以加大  $\text{win}(r_{\text{win}}(t))$ , 进一步扩展机器人的探测范围, 寻找更优的子目标; 当  $\varphi_F$  较小时, 应该减小  $\text{win}(r_{\text{win}}(t))$ , 缩小机器人的探测范围, 以便在狭窄空间中找到可行路径。即

$$r_{\text{win}}(t+1) = r_{\text{win}}(t) + \Delta r,$$

式中  $\Delta r$  为  $\varphi_F$  的函数, 当  $0 \leq \varphi_F < \varphi_{F,1}$  时,  $\Delta r = \Delta r_{\min}$ ; 当  $\varphi_{F,1} \leq \varphi_F < \varphi_{F,2}$  时,  $\Delta r = (\Delta r_{\max} - \Delta r_{\min})(\varphi_F - \varphi_{F,2})/(\varphi_{F,2} - \varphi_{F,1}) + \Delta r_{\max}$ ; 当  $\varphi_{F,2} \leq \varphi_F < \varphi_{F,\max}$  时,  $\Delta r = \Delta r_{\max}$ , 其中,  $\varphi_{F,1}$  和  $\varphi_{F,2}$  为设置的常数,  $\varphi_{F,\max}$  为最大值,  $\Delta r_{\min}$  和  $\Delta r_{\max}$  为定义的最大和最小值。

2.3 机器人的沿墙行走行为

2.3.1 局部极小的判断方法

当机器人陷入局部极小时, 机器人路径轨迹会出现振荡, 或沿着某一闭合路径运动, 从而不能到达目标点。为了避免这种情况, 需要提前预测机器人是否会进入局部极小状态, 通过角度比较条件  $|\theta_{\text{goal}} - \theta_R| \geq 90^\circ$  来判断, 若条件满足, 且目标点位于机器人的左侧, 则 follow-left-wall (沿左墙) 行为被激活; 若条件满足但目标点位于机器人的右侧, 则 follow-right-wall (沿右墙) 行为被激活。用 WF 解死锁算法使机器人沿着障碍物的边界运动, 直到满足退出条件  $|\theta_{\text{goal}} - \theta_R| < 90^\circ$ 。

## 2.3.2 即时子目标的确定

如图 3 所示,当机器人选择 follow-left-wall

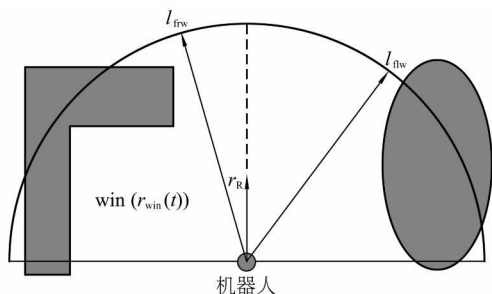


图 3 沿墙行走行为示意图

行为时,采用  $\text{win}(r_{\text{win}}(t))$  中左起第一个障碍物的右边界线扩展线对应的扫描点为子目标,即图中的  $l_{\text{fw}}$ ;当选择 follow-right-wall 行为时,采用  $\text{win}(r_{\text{win}}(t))$  中右起第一个障碍物的左边界线扩展线对应的扫描点为子目标,即图中的  $l_{\text{rw}}$ . 机器人对确定的子目标点进行跟踪,即可沿着障碍物行走;若机器人沿障碍绕行的距离超过设定值  $L_{\text{max}1}$  时仍未退出,则自转  $180^\circ$  再向另一沿墙走方式运动;在新的沿墙走方式下,若绕行的距离超过设定值  $L_{\text{max}2}$  仍未退出(取  $L_{\text{max}2} > 2L_{\text{max}1}$ ),则停止

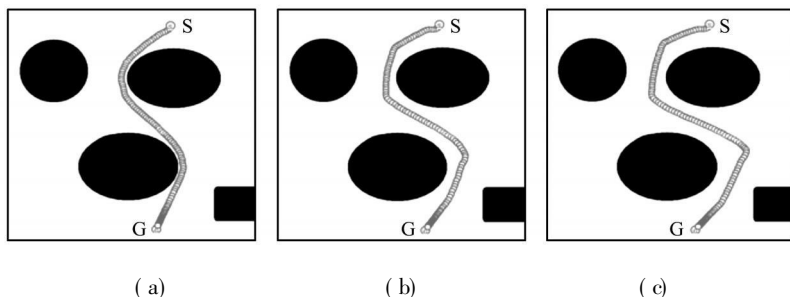


图 4 简单环境中机器人的路径规划

障碍物的距离却最近,这种情况很不利于机器人的安全行驶;图 4(b) 为  $k_{\text{gh}} = 0.70$ ,  $k_{\text{safe}} = 0.30$ ,  $k_{\text{smooth}} = 0$  时的路径,考虑了安全性,与图 4(a) 相比安全性明显得到提高(路径远离障碍物);图 4(c) 为  $k_{\text{gh}} = 0.55$ ,  $k_{\text{safe}} = 0.20$ ,  $k_{\text{smooth}} = 0.25$  时的路径,进一步考虑了运动平稳性,有利于减少机器人的转向操作。

当障碍物形状过于复杂或特殊时,采用滚动窗算法,机器人单靠实时探测的局部环境知识无法进一步规划路径,就会陷入死锁或振荡状态<sup>[4]</sup>. 同文献[4]地图及起始点、目标点下,本文算法借助有效的 WF 行为可以脱离局部极小而顺利到目标点. 在文献[1~5]算法中,滚动窗口过小时,机器人无法发现更优的行走方向<sup>[4]</sup>;在障碍物比较密集的环境,当窗口太大时,将出现窗口边界点均为禁入点而找不到局部目标点的情况. 文献

路径(认为目标不可达). 在 WF 行为中(follow-left-wall 或 follow-right-wall),若发现目标点  $P_G$  位于  $\text{win}(r_{\text{win}}(t))$  的可通行域中,则直接以  $P_G$  作为当前的即时子目标.

## 2.3.3 规划窗口的动态调整

在 WF 行为中,若为 follow-left-wall,且  $f_{l_{181}}^{\text{trav}} = 0$ ,则设置  $r_{\text{win}}(t+1) = d_{181}$ ,否则窗口大小不变;若为 follow-right-wall,且  $f_{l_1}^{\text{trav}} = 0$ ,则设置  $r_{\text{win}}(t+1) = d_1$ ,否则窗口大小不变.

## 3 仿真实验结果及分析

以圆形表示移动机器人,半径  $r_R = 0.3 \text{ m}$ ,安全半径取  $r_{\text{saf}} = 0.7 \text{ m}$ . 机器人的动力学约束为:线速度  $v_{R\text{max}} = 1.0 \text{ m/s}$ ,  $a_{R\text{max}} = 0.4 \text{ m/s}^2$ ,最大不打滑负加速度  $a_{N\text{max}} = -1.2 \text{ m/s}^2$ ;角速度  $\omega_{R\text{max}} = 2.0 \text{ rad/s}$ ,  $\theta_{R\text{max}} = 0.5 \text{ rad/s}^2$ .

为了验证多约束环境模型中各种因素对路径规划的影响,图 4 给出简单环境下的路径规划结果. 图 4(a) 为  $k_{\text{gh}} = 1$ ,  $k_{\text{safe}} = k_{\text{smooth}} = 0$  时的路径,只考虑了可通行性,所以尽管其路径最短,但离障

[9~11]算法的窗口太大,即“曲带”太长时,也会出现这种问题. 因此,这种基于局部窗口的滚动规划算法中,若窗口太小,则机器人不能及时提前进行避障行走;若窗口太大,则机器人可能找不到可选子目标. 当采用基于自适应窗口的路径规划方法时,机器人能够根据环境信息对局部规划窗口的大小进行动态调整,所得路径如图 5(a) 所示,其窗口大小的调整过程如图 5(b) 所示. 可以看出,图 5(a) 中路径优于文献[9~11]的路径,而且由本文中对算法的描述可知,机器人路径规划的计算量与窗口大小无关,不会随着窗口的增大而增大每一次滚动规划中的局部规划时间.

仿真结果表明,本文基于环境建模的实时路径规划方法具有很好的环境适应性和避障能力,获得了较好的安全性和可达性,并具有较高的计算实时性,适合于机器人的实际应用.

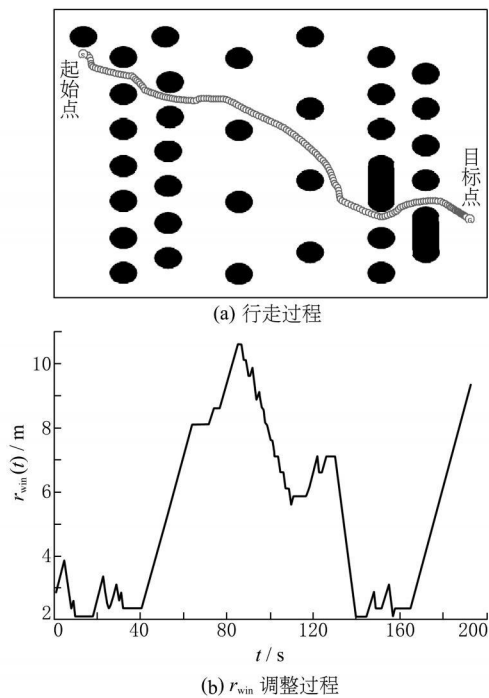


图5 窗口大小自适应调整

## 参 考 文 献

- [1] 张纯刚, 席裕庚. 全局环境未知时基于滚动窗口的机器人路径规划[J]. 中国科学: E 辑, 2001, 31(1): 51-58.
- [2] 席裕庚, 张纯刚. 一类动态不确定环境下机器人的滚动路径规划[J]. 自动化学报, 2002, 28(2): 161-175.
- [3] 张纯刚, 席裕庚. 动态未知环境中移动机器人的滚动路径规划及安全性分析[J]. 控制理论与应用, 2003,

20(1): 37-44.

- [4] 张纯刚, 席裕庚. 机器人滚动路径规划的算法与仿真研究[J]. 高技术通讯, 2003(4): 53-57.
- [5] 杜军君, 席裕庚. 一种改进的机器人滚动路径规划算法[J]. 控制工程, 2006, 13(S): 183-204.
- [6] Ge S S, Lai Xuecheng, Mamun A A. Sensor-based path planning for nonholonomic mobile robots subject to dynamic constraints[J]. Robotics and Autonomous Systems, 2007, 55(7): 513-526.
- [7] Lai Xuecheng, Ge S S, Mamun A A I. Hierarchical incremental path planning and situation-dependent optimized dynamic motion planning considering accelerations[J]. IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics, 2007, 37(6): 1541-1554.
- [8] Blanco J L, Gonzalez J, Fernandez-Madruga J A. Extending obstacle avoidance methods through multiple parameter-space transformations[J]. Autonomous Robots, 2008, 24(1): 29-48.
- [9] Kamon I, Rivlin E, Rimon E. A new range-sensor based globally convergent navigation algorithm for mobile robots[C]// IEEE International Conference on Robotics and Automation. Minneapolis: IEEE, 1996: 429-435.
- [10] Kamon I, Rivlin E. Sensory-based motion planning with global proofs[J]. IEEE Transactions on Robotics and Automation, 1997, 13(6): 814-822.
- [11] 郝宗波, 洪炳熔. 未知环境下基于传感器的移动机器人路径规划[J]. 电子学报, 2006, 34(5): 953-956.